GPU Acceleration of Particle-based Volume Rendering using CUDA

Ding Zhongming[1)], Naohisa Sakamoto[2),3)], Yasuo Ebara[4)], Koji Koyamada[3)]
1) Graduate School of Electrical Engineering, Kyoto University
2) Center for the Promotion of Excellence in Higher Education, Kyoto University
3) KGT inc.
4) Academic Center for Computing and Media Studies, Kyoto University

ABSTRACT: In this paper, we apply Particle-based Volume Rendering (PBVR) technique using a current programmable GPU architecture. Recently, the increasing programmability of GPU offers an efficient method of SIMD parallel algorithm to solve the speed problem. Due to the each point or pixel can be calculated independently, we use programmable graphics hardware to delegate all expensive rendering tasks to the GPU. Here we apply on the popular programming architecture CUDA based on GeForce 8800 graphics unit. This approach allows enormous volume particles to be rendered in SIMD way instead of time-costing sequence processing so that the rendering speed can be accelerated. In this processing, each particle can be handled separately by one of the multi-processors in GPU. In this implementation, we introduce a non-confliction way to map the conventional algorithm onto CUDA calculation architecture efficiently. We apply CUDA in the programming framework as a general purpose GPU calculation for PBVR instead of using conventional GPU pipeline. All the rendering flow can be divided into three stages: the beforehand data arrangement, particle projection and sub-pixel processing. In order to evaluate the performance, we compare the frame rate of GPU accelerated PBVR with traditional CPU based approach.

KEY WORDS: GPU based volume rendering, particle-based volume rendering, CUDA

1. INTRODUCTION

Particle-based approaches are widely used in 3D scan data display and large scale particles visualization. Since the scale and complexity of volume datasets is increasing, there is a strong demand for volume rendering techniques capable of handling huge and complex volume datasets. To render such volume datasets with conventional volume rendering techniques, the calculations required for sorting and alpha blending can become a bottleneck.

Particle-based volume rendering (PBVR) technique can be effectively utilized to render various types of volume datasets, either regular or irregular comparing against ray-casting method [1]. In order to gain a proper rendering quality, large amount of particles need to be generated and projected for pixel processing. In the other hand, high quality result image can also be acquired by enhancing the sub-pixel level. However either of two stages costs large CPU processing time.

Recently, the increasing programmability of GPU offers an efficient method for SIMD parallel algorithm designing. Due to the each point or pixel can be calculated independently, we use programmable graphics hardware we delegate all expensive rendering tasks to the GPU. Here we implement on the popular programming architecture CUDA based on GeForce 8800 graphics unit.

The time cost of large scale data computing is always come up with heavy data access time waste. In the case of GPU based rendering, huge amounts of raw data transferring via different memory devices usually causes severe bottle neck during the processing flow. Since the local data processing speed of GPU multiprocessor is relatively faster than memory data fetch, designing an appropriate memory data arrangement become a main focus of our work.

Usually the SIMD parallel computing often brings a risk of memory access confliction which could drastically cumber the processing or even causes a crash. In this paper we also discuss about a forehand predication sorting method to avoid confliction which is likely to happen in buffer writing.

2. OVERVIEW OF PBVR

Recently, volume rendering technique which uses particles as rendering primitives has received increasing attention and is a topic of ongoing research. Hopf and Ertl proposed the use of a hierarchical data structure in order to accelerate the visualization of scattered particle data [1]. This data structure enables fast sorting of semi-transparent clusters and may therefore trade rendering speed for image quality. The image is obtained by traversing this structure for each frame. In addition, the use of quantized relative coordinates reduced the memory consumption. In order to realize a semi-transparent effect, the technique requires alpha blending which requires the particles to be sorted according to their projected z-coordinates.

The proposed particle-based volume rendering technique represents the volume data as a set of particles. The particle density is derived from a user-specified transfer function and describes the probability that a particle is present at a given position. Since these